Rob-GAN: Generator, Discriminator, and Adversarial Attacker

Xuanqing Liu Cho-Jui Hsieh
University of California, Los Angeles
{xqliu, chohsieh}@cs.ucla.edu

Abstract

We study two important concepts in adversarial deep learning—adversarial training and generative adversarial network (GAN). Adversarial training is the technique used to improve the robustness of discriminator by combining **adversarial attacker** *and* **discriminator** *in the training phase. GAN is commonly used for image generation by jointly optimizing* **discriminator** *and* **generator**. *We show these two concepts are indeed closely related and can be used to strengthen each other—adding a generator to the adversarial training procedure can improve the robustness of discriminators, and adding an adversarial attack to GAN training can improve the convergence speed and lead to better generators. Combining these two insights, we develop a framework called Rob-GAN to jointly optimize generator and discriminator in the presence of adversarial attacks—the generator generates fake images to fool discriminator; the adversarial attacker perturbs real images to fool discriminator, and the discriminator wants to minimize loss under fake and adversarial images. Through this end-to-end training procedure, we are able to simultaneously improve the convergence speed of GAN training, the quality of synthetic images, and the robustness of discriminator under strong adversarial attacks. Experimental results demonstrate that the obtained classifier is more robust than state-of-the-art adversarial training approach [23], and the generator outperforms SN-GAN on ImageNet-143.*

1. Introduction

Adversarial deep learning has received a significant amount of attention in the last few years. In this paper, we study two important but different concepts—adversarial attack/defense and generative adversarial network (GAN). Adversarial attacks are algorithms that find a highly resembled images to cheat a classifier. Training classifiers under adversarial attack (also known as adversarial training) has become one of the most promising ways to improve the robustness of classifiers [23]. On the other hand, GAN is a generative model where the generator learns to convert white noise to images that look authentic to the discriminator [11, 28]. We show in this paper that they are indeed closely related and can be used to strengthen each other, specifically we have the following key insights:

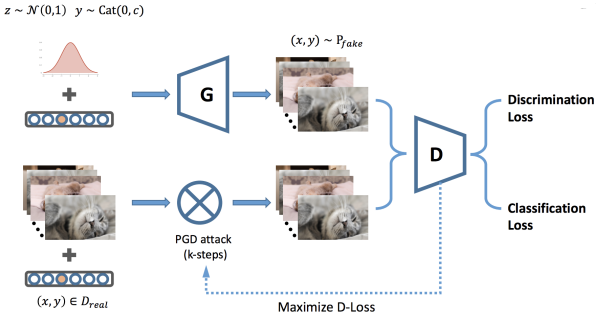

Figure 1: Illustration of the training process. This is similar to the standard GAN training, i.e. alternatively updating the generator G and discriminator D networks. The main difference is that whenever feeding the real images to the D network, we first invoke adversarial attack, so the discriminator is trained with adversarial examples.

1. The robustness of adversarial trained classifier can be improved if we have a deeper understanding of the image distribution. Therefore a generator can improve the adversarial training process.
2. GAN training can be very slow to reach the equilibrium if the discriminator has a large curvature on the image manifold. Therefore an adversarial trained discriminator can accelerate GAN training.

Based on these findings, we managed to accelerate and stabilize the GAN training cycle, by enforcing the discriminator to stay robust on image manifold. At the same time, since data augmentation is used in the robust training process, the generator provides more information about the data distribution. Therefore we get a more robust classifier that generalizes better to unseen data. Our contributions can be summarized as follows:

1. We give insights why the current adversarial training algorithm does not generalize well to unseen data. Parallelly, we explain why the GAN training is slow to

*Project repository:

reach an equilibrium.

2. We draw a connection between adversarial training and GAN training, showing how they can benefit each other: we can use GAN to improve the generalizability of adversarial training, and use adversarial training to accelerate GAN training and meanwhile make it converge to a better generator.
3. We propose a novel framework called `Rob-GAN`, which integrates generator, discriminator and adversarial attacker as a three-player game. And we also show how to train the framework efficiently in an end to end manner.
4. We formulate a better training loss for conditional GAN by reformulating the AC-GAN loss.
5. We design a series of experiments to confirm all the hypotheses and innovations made in the text. For example, with GAN data augmentation, we can improve the accuracy of state-of-the-art adversarial training method [23] from 29.6% to 36.4% on ResNet18(+CIFAR10) under a strong adversarial attack. Moreover, we observe a $3 \sim 7$x speedup in terms of convergence rate, when inserting the adversarial attacker into GAN training cycle. Lastly, our model attains better inception scores on both datasets, compared with the strong baseline (SN-GAN [26]).

Notations Throughout this paper, we denote the (image, label) pair as (x_i, y_i), i is the index of data point; The classifier parameterized by weights w is $f(x; w)$, this function includes the final `Softmax` layer so the output is probabilities. Loss function is denoted as $\ell(\cdot, \cdot)$. We also define $D(x)$ and $G(z)$ as the discriminator and generator networks respectively. The adversarial example x_{adv} is crafted by perturbing the original input, i.e. $x_{\text{adv}} = x + \delta$, where $\|\delta\| \leq \delta_{\max}$. For convenience, we consider ℓ_∞-norm in our experiments. The real and fake images are denoted as $x_{\text{real/fake}}$. Note that in this paper "fake" images and "adversarial" images are different: fake images are generated by generator, while adversarial images are made by perturbing the natural images with small (carefully designed) noise. The training set is denoted as $\mathcal{D}_{\text{tr}}$, with N_{tr} data points. This is also the empirical distribution. The unknown data distribution is $\mathcal{P}_{\text{data}}$. Given the training set $\mathcal{D}_{\text{tr}}$, we define empirical loss function $\frac{1}{N_{\text{tr}}} \sum_{i=1}^{N_{\text{tr}}} \ell(f(x_i; w), y_i) = \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{tr}}} \ell(f(x; w), y)$.

2. Background and Related Work

2.1. Generative Adaversarial Network

A GAN has two competing networks with different objectives: in the training phase, the generator $G(z)$ and the discriminator $D(x)$ are evolved in a minimax game, which can be denoted as a unified loss:

$$\min_G \max_D \Big\{ \mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\text{tr}}} \big[\log D(x)\big] + \mathop{\mathbb{E}}_{z\sim\mathcal{P}_z} \big[\log(1 - D(G(z)))\big] \Big\}, \tag{1}$$

where $\mathcal{P}_z$ is the distribution of noise. Unlike traditional machine learning problems where we typically minimize the loss, (1) is harder to optimize and that is the focus of recent literature. Among them, a guideline for the architectures of G and D is summarized in [30]. For high resolution and photo-realistic image generation, currently the standard way is to first learn to generate low resolution images as the intermediate products, and then learn to refine them progressively [9, 20]. This turns out to be more stable than directly generating high resolution images through a gigantic network. To reach the equilibrium efficiently, alternative loss functions [1, 2, 5, 13, 37] are applied and proven to be effective. Among them, [1] theoretically explains why training DCGAN is highly unstable. Following that work, [2] proposes to use Wasserstein-1 distance to measure the distance between real and fake data distribution. The resulting network, namely "Wasserstein-GAN", largely improves the stability of GAN training. Another noteworthy work inspired by WGAN/WGAN-GP is spectral normalization [26]. The main idea is to estimate the operator norm $\sigma_{\max}(W)$ of weights W inside layers (convolution, linear, etc.), and then normalize these weights to have 1-operator norm. Because ReLU non-linearity is 1-Lipschitz, if we stack these layers together the whole network will still be 1-Lipschitz, which is exactly the prerequisite to apply Kantorovich-Rubinstein duality to estimate Wasserstein distance.

2.2. Adversarial attacks and defenses

Another key ingredient of our method is adversarial training, originated in [35] and further studied in [12]. They found that machine learning models can be easily "fooled" by slightly modified images if we design a tiny perturbation according to some "attack" algorithms. In this paper we apply a standard algorithm called PGD-attack [23] to generate adversarial examples. Given an example x with ground truth label y, PGD computes adversarial perturbation δ by solving the following optimization with Projected Gradient Descent:

$$\delta \coloneqq \mathop{\arg\max}_{\|\delta\|\leq\delta_{\max}} \ell\big(f(x+\delta; w), y\big), \tag{2}$$

where $f(\cdot; w)$ is the network parameterized by weights w, $\ell(\cdot, \cdot)$ is the loss function and for convenience we choose $\|\cdot\|$ to be the ℓ_∞-norm in accordance with [23, 4], but note that other norms are also applicable. Intuitively, the idea of (2) is to find the point $x_{\text{adv}} \coloneqq x + \delta$ within an ℓ_∞-ball such that the loss value of x_{adv} is maximized, so that point is most likely to be an adversarial example. In fact, most optimization-based attacking algorithms (e.g. FGSM [12], C&W [7]) share the same idea as PGD attack.

Opposite to the adversarial attacks, the adversarial defenses are techniques that make models resistant to adversarial examples. It is worth noting that defense is a much harder task compared with attacks, especially for

high dimensional data combined with complex models. Despite that huge amount of defense methods are proposed [29, 23, 6, 22, 15, 10, 39, 34, 31], which can be identified as either random based, projection based, or de-noiser based. In the important overview paper [4, 3], adversarial training [23] is acknowledged as one of the most powerful defense algorithm, which can be formulated as

$$\min_{w} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}_{\text{data}}} \Big[\max_{\|\delta\|\le\delta_{\max}} \ell\big(f(x+\delta;w),y\big)\Big], \qquad (3)$$

where $(x, y) \sim \mathcal{P}_{\text{data}}$ is the (image, label) joint distribution of data, $f(x; w)$ is the network parameterized by w, $\ell\big(f(x; w), y\big)$ is the loss function of network (such as the cross-entropy loss). We remark that the ground truth data distribution $\mathcal{P}_{\text{data}}$ is not known in practice, which will be replaced by the empirical distribution.

It is worth noting that one of our contributions is to use GAN to defend the adversarial attacks, which is superficially similar to Defense-GAN [32]. However, they are totally different underneath: the idea of Defense-GAN is to project an adversarial example to the space of fake images by minimizing ℓ_2 distance: $x_{\text{out}} = \arg\min_{G(z)} \|x^{\text{adv}} - G(z)\|^2$, and then making prediction on the output x_{out}. In contrast, our defense mechanism is largely based on adversarial training [23]. Another less related work that applies GAN in adversarial setting is AdvGAN [38], where GAN is used to generate adversarial examples.

3. Proposed Approach

We propose a framework called `Rob-GAN` to jointly optimize generator and discriminator in the presence of adversarial attacks—the generator generates fake images to fool discriminator; the adversarial attack perturbs real images to fool discriminator, and the discriminator wants to minimize loss under fake and adversarial images (see Fig. 1). In fact, `Rob-GAN` is closely related to both adversarial training and GAN. If we remove generator, `Rob-GAN` becomes standard adversarial training method. If we remove adversarial attack, `Rob-GAN` becomes standard GAN. But why do we want to put these three components together? Before delving into details, we first present two important motivations: I) Why can GAN improve the robustness of adversarial trained discriminator? II) Why can adversarial attacker improve the training of GAN?

We answer I) and II) in Section 3.1 and 3.2, and then give details of `Rob-GAN` in Section 3.3.

3.1. Insight I: The generalization gap of adversarial training — GAN aided adversarial training

In Sec. 2.2 we listed some works on adversarial defense, and pointed out that adversarial training is one of the most effective defense method to date. However, until now this method has only been tested on small dataset like MNIST and CIFAR10 and it is still an open problem whether adversarial training can scale to large dataset such as ImageNet. Furthermore, although adversarial training leads to certified robustness on training set (due to the design of the objective function (3)), the performance usually drops significantly on the test set. This means that **the generalization gap is large** under adversarial attacks (Fig. 2 (*Left*)). In other words, despite that it is hard to find an adversarial example near the training data, it is much easier to find one near the testing data. In the following, we investigate the reason behind this huge (and enlarging) generalization gap, and later we will solve this problem with GAN aided adversarial training.

From statistical learning theory, it is known that the generalization ability of model relies on the convergence of empirical risk to population risk, formally:

$$\sup_{h\in\mathcal{F}} \Big| \frac{1}{n}\sum_{i=1}^{n} h(X_i) - \mathbb{E}_X[h(X)] \Big| \xrightarrow{a.s.} 0, \text{ when } n\to\infty, \qquad (4)$$

where $\mathcal{F}$ is the set of hypotheses that are assumed to be L-Lipschitz continuous and X can be any sub-Gaussian random variable. Furthermore, to make our model robust to adversarial distortion, it is desirable to enforce a small local Lipschitz value (LLV) on the underlining data distribution $\mathcal{P}_{\text{data}}$. This idea includes many of the defense methods such as [8]. In essence, restricting the LLV can be formulated as a composite loss minimization problem:

$$\min_{w} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}_{\text{data}}} \Big[\ell\big(f(x;w),y\big) + \lambda \cdot \big\| \frac{\partial}{\partial x} \ell\big(f(x;w),y\big) \big\|_2 \Big]. \qquad (5)$$

Note that (5) can be regarded as the "linear expansion" of (3). In practice we do not know the ground truth data distribution $\mathcal{P}_{\text{data}}$; instead, we use the empirical distribution to replace (5):

$$\min_{w} \frac{1}{N_{\text{tr}}} \sum_{i=1}^{N_{\text{tr}}} \Big[\ell\big(f(x_i;w),y_i\big) + \lambda \cdot \big\| \frac{\partial}{\partial x_i} \ell\big(f(x_i;w),y_i\big) \big\|_2 \Big], \qquad (6)$$

where $\{(x_i, y_i)\}_{i=1}^{N_{\text{tr}}}$ are feature-label pairs of the training set. Ideally, if we have enough data and the hypotheses set is moderately large, the objective function in (6) still converges to (5). However when considering adversarial robustness, we have one more problem to worry about:

> *Does small LLV in training set automatically generalize to test set?*

The enlarged accuracy gap shown in Fig. 2 (*Left*) implies a negative answer. To verify this phenomenon in an explicit way, we calculate Lipschitz values on samples from training and testing set separately (Fig. 2 (*Right*)). We can see that similar to the accuracy gap, the LLV gap between

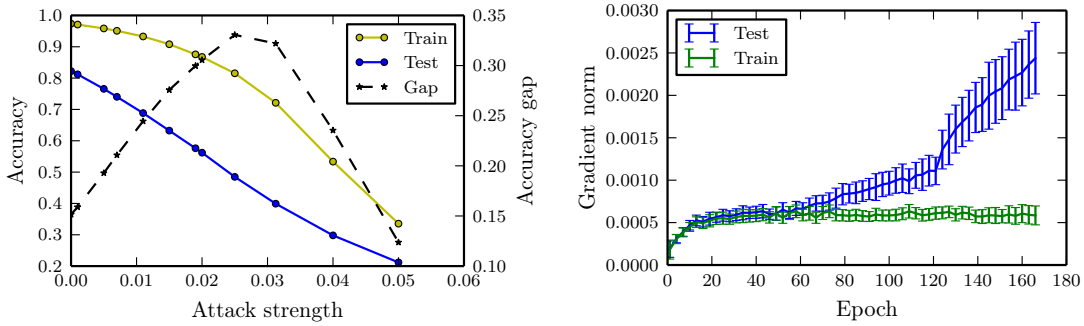

Figure 2: *Left*: Accuracy under different levels of attack. The model (VGG16) is obtained by adversarial training on CIFAR-10 with the maximum perturbation in adversarial training set as $8/256$. We can observe that: 1) the accuracy gap between training and testing is large, 2) the gap is even larger than the attack strength (after attack strength ≈ 0.03, both training and testing accuracy go down to zero, so the gap also decreases). *Right*: The local Lipschitz value (LLV) measured by gradient norm $\|\frac{\partial}{\partial x_i}\ell\big(f(x_i;w),y_i\big)\|_2$. Data pairs (x_i, y_i) are sampled from the training and testing set respectively. During the training process, LLV on the training set stabilizes at a low level, while LLV on the test set keeps growing.

training and testing set is equally large. Thus we conclude that **although adversarial training controls LLV around training set effectively, this property does not generalize to test set**. Notice that our empirical findings do not contradict the certified robustness of adversarial training using generalization theory (e.g. [33]), which can be loose when dealing with deep neural networks.

The generalization gap can be reduced if we have a direct access to the whole distribution $\mathcal{P}_{\text{data}}$, instead of approximating it by limited training data. This leads to our first motivation:

> *Can we use GAN to learn $\mathcal{P}_{data}$ and then perform the adversarial training process on the learned distribution?*

If so, then it becomes straightforward to train an even more robust classifier. Here we give the loss function for doing that, which can be regarded as composite robust optimization on both original training data and GAN synthesized data:

$$\begin{aligned}
&\min_w \mathcal{L}_{\text{real}}(w,\delta_{\max}) + \lambda\cdot\mathcal{L}_{\text{fake}}(w,\delta_{\max}),\\
&\mathcal{L}_{\text{real}}(w,\delta_{\max}) \triangleq \frac{1}{N_{\text{tr}}}\sum_{i=1}^{N_{\text{tr}}}\max_{\|\delta_i\|\le\delta_{\max}}\ell\big(f(x_i+\delta_i;w);y_i\big),\\
&\mathcal{L}_{\text{fake}}(w,\delta_{\max}) \triangleq \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}_{\text{fake}}}\max_{\|\delta\|\le\delta_{\max}}\ell\big(f(x+\delta;w);y\big).
\end{aligned} \tag{7}$$

Again the coefficient λ is used to balance the two losses. To optimize the objective function (7), we adopt the same stochastic optimization algorithm as adversarial training. That is, at each iteration we draw samples from either training or synthesized data, find the adversarial examples, and then calculate stochastic gradients upon the adversarial examples. We will show the experimental results in Sec. 4.

3.2. Insight II: Accelerate GAN training by robust discriminator

If even a well trained deep classifier can be easily "cheated" by adversarial examples, so can the others. Recall in conditional GANs, such as AC-GAN [28], the discriminator should not only classify real/fake images but also assign correct labels to input images. Chances are that, if the discriminator is not robust enough to the adversarial attacks, then the generator could make use of its weakness and "cheat" the discriminator in a similar way. Furthermore, even though the discriminator can be trained to recognize certain adversarial patterns, the generator will find out other adversarial patterns easily, so the minimax game never stops. Thus we make the following hypothesis:

Fast GAN training relies on robust discriminator.

Before we support this hypothesis with experiments, we briefly review the development of GANs: the first version of GAN objective [11] is unstable to train, WGAN [2, 14] adds a gradient regularizer to enforce the discriminator to be globally 1-Lipschitz continuous. Later on, SN-GAN [26] improves WGAN by replacing gradient regularizer with spectral normalization, again enforcing 1-Lipschitz continuity globally in discriminator. We see both methods *implicitly* make discriminator to be robust against adversarial attacks, because a small Lipschitz value (e.g. 1-Lipschitz) enables stronger invariance to adversarial perturbations.

Despite the success along this line of research, we wonder if a weaker but smarter regularization to the discriminator is possible. After all, if the regularization effect is too strong, then the model expressiveness will be restricted. Concretely, instead of a strict **one**-Lipschitz function **globally**, we require a **small local** Lipschitz value on image manifold. As we will see, this can be done conveniently through adversarial training to the discriminator. In this way, we can draw a

connection between the robustness of discriminator and the learning efficiency of generator, as illustrated in Fig. 3.

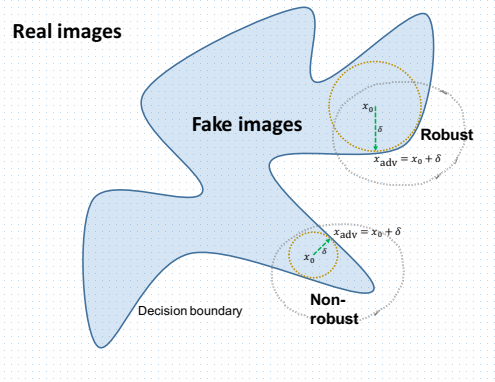

Figure 3: Comparing robust and non-robust discriminators, for simplicity, we put them together into one graph. Conceptually, the non-robust discriminator tends to make all images close to the decision boundary, so even a tiny distortion δ can move a fake image x_0 to across the decision boundary and leads to a mis-classification: $x_{\text{adv}} = x_0 + \delta$. In contrast, such δ is expected to be much larger for robust discriminators.

As one can see in Fig. 3, if a discriminator $D(x)$ has small LLV (equivalently, small $\|D'(x)\|$), then we know $D(x + \delta) \approx D(x) + D'(x) \cdot \delta \approx D(x)$ for a "reasonably" small δ. In other words, for a robust discriminator, the perturbed fake image $x_{\text{adv}} = x_0 + \delta$ is unlikely to be misclassified as real image, unless δ is large. Different from the setting of adversarial attacks (2), in GAN training, the "attacker" is now a generator network $G(z; w)$ parameterized by $w \in \mathbb{R}^d$. Suppose at time t, the discriminator can successfully identify fake images, or equivalently $D(G(z; w^t)) \approx 0$ for all z, then at time $t + 1$ what should the generator do to make $D(G(z; w^{t+1})) \approx 1$? We can develop the following bound by assuming the Lipschitz continuity of $D(x)$ and $G(z; w)$,

$$\begin{aligned} 1 \approx & D(G(z; w^{t+1})) - D(G(z; w^t)) \\ \lessapprox & \|D'\big(G(z; w^t)\big)\| \cdot \|G(z; w^{t+1}) - G(z; w^t)\| \\ \lessapprox & \|D'\big(G(z; w^t)\big)\| \cdot \|\frac{\partial}{\partial w} G(z; w^t)\| \cdot \|w^{t+1} - w^t\| \\ \leq & L_D L_G \|w^{t+1} - w^t\|, \end{aligned} \tag{8}$$

where $L_{D,G}$ indicates the Lipschitz constants of discriminator and generator. As we can see, the update of generator weights is inversely proportional to L_D and L_G: $\|w^{t+1} - w^t\| \propto \frac{1}{L_D L_G}$. If the discriminator is lacking robustness, meaning L_D is large, then the generator only needs to make a small movement from the previous weights w^t, making the convergence very slow. This validates our hypothesis that *fast GAN training relies on robust discriminator*. In the experiment section, we observe the same phenomenon in all two experiments, providing a solid support for this hypothesis.

3.3. `Rob-GAN`: Adversarial training on learned image manifold

Motivated by Sec. 3.1 and 3.2, we propose a system that combines generator, discriminator, and adversarial attacker into a single framework. Within this framework, we conduct end-to-end training for both generator and discriminator: the generator feeds fake images to the discriminator; meanwhile real images sampled from training set are preprocessed by PGD attacking algorithm before sending to the discriminator. The network structure is illustrated in Fig. 1.

Discriminator and the new loss function: The discriminator could have the standard architecture like AC-GAN. At each iteration, it discriminates real and fake images. When the ground truth labels are available, it also predicts the classes. In this paper, we only consider the conditional GANs proposed in [25, 28, 27], and their architectural differences are illustrated in Fig. 4. Among them we simply choose AC-GAN, despite that SN-GAN (a combination of spectral normalization [26] and projection discriminator [27]) performs much better in their paper. The reason we choose AC-GAN is that SN-GAN's discriminator relies on the ground truth labels and their objective function is not designed to encourage high classification accuracy. But surprisingly, even though AC-GAN is beaten by SN-GAN by a large margin, after inserting the adversarial training module, the performance of AC-GAN matches or even surpasses the SN-GAN, due to the reason discussed in Sec. 3.2.

We also improved the loss function of AC-GAN. Recall that the original loss in [28] defined by discrimination likelihood $\mathcal{L}_S$ and classification likelihood $\mathcal{L}_C$:

$$\begin{aligned} \mathcal{L}_S &= \mathbb{E}[\log \mathbb{P}(S = \text{real}|X_{\text{real}})] + \mathbb{E}[\log \mathbb{P}(S = \text{fake}|X_{\text{fake}})] \\ \mathcal{L}_C &= \mathbb{E}[\log \mathbb{P}(C = c|X_{\text{real}})] + \mathbb{E}[\log \mathbb{P}(C = c|X_{\text{fake}})], \end{aligned} \tag{9}$$

where $X_{\text{real/fake}}$ are any real/fake images, S is the discriminator output, and C is the classifier output. Based on (9), the goal of discriminator is to maximize $\mathcal{L}_S + \mathcal{L}_C$ while generator aims at maximizing $\mathcal{L}_C - \mathcal{L}_S$. According to this formula, both G and D are trained to increase $\mathcal{L}_C$, which is problematic because even if $G(z; w)$ generates bad images, $D(x)$ has to struggle to classify them (with high loss), and in such case the corresponding gradient term $\nabla \mathcal{L}_C$ can contribute uninformative direction to the discriminator. To resolve this issue, we split $\mathcal{L}_C$ to separate the contributions of real and fake images,

$$\begin{aligned} \mathcal{L}_{C_1} &= \mathbb{E}[\log \mathbb{P}(C = c|X_{\text{real}})] \\ \mathcal{L}_{C_2} &= \mathbb{E}[\log \mathbb{P}(C = c|X_{\text{fake}})], \end{aligned} \tag{10}$$

then discriminator maximizes $\mathcal{L}_S + \mathcal{L}_{C_1}$ and generator maximizes $\mathcal{L}_{C_2} - \mathcal{L}_S$. The new objective function ensures that discriminator only focuses on classifying real images and discriminating real/fake images, and the classifier branch will not be distracted by the fake images.

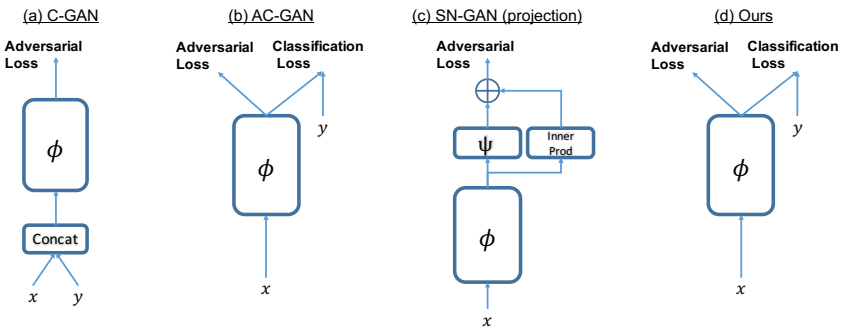

Figure 4: Comparing the architectures of discriminators. Our architecture is similar to AC-GAN [28], but they are different in loss functions, if one compares (9) with (10). (x, y) is (image, label) pair, ϕ and ψ denote different network blocks. Recall in AC-GAN and our architecture, the discriminator has two branches, one is for discriminating "real/fake" images and the other is for classification.

Generator: Similar to the traditional GAN training, the generator is updated on a regular basis to mimic the distribution of real data. This is the key ingredient to improve the robustness of classification task: as shown in Sec. 3.1, model from adversarial training performs well on training set but is vulnerable on test set. Intuitively, this is because during adversarial training, the network only "sees" adversarial examples residing in the small region of all training samples, whereas the rest images in the data manifold are undefended. Data augmentation is a natural way to resolve this issue, but traditional data augmentation methods like image jittering, random resizing, rotation, etc. [21, 16, 36, 40, 18] are all simple geometric transforms, they are useful but not effective enough: even after random transforms, the total number of training data is still much fewer than required. Instead, our system has unlimited samples from generator to provide a continuously supported probability density function for the adversarial training. Unlike traditional augmentation methods, if the equilibrium in (1) is reached, then we can show that the solution of (1) would be $\mathcal{P}_{\text{fake}}(z) \overset{\text{dist.}}{=} \mathcal{P}_{\text{real}}$ [11], and therefore the classifier can be trained on the ground truth distribution $\mathcal{P}_{\text{real}}$.

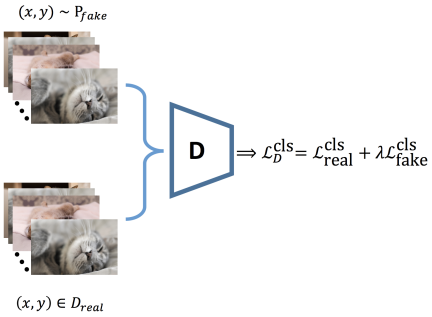

Figure 5: Illustration of fine-tuning the discriminator. We omit the adversarial attack here for brevity.

Fine-tuning the classifier: After end-to-end training, the discriminator has learned to minimize both **discrimination loss** and **classification loss** (see Fig. 1). If we want to train the discriminator to conduct a pure multi-class classification task, we will need to fine-tune it by combining fake and real images and conducting several steps of SGD only on the robustness classification loss (illustrated in Fig. 5):

$$\begin{aligned}\mathcal{L}_D^{\text{cls}} \triangleq & \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}_{\text{real}}} \ell(f(x_{\text{adv}}; w), y) + \\ & \lambda \cdot \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}_{\text{fake}}} \ell(f(x_{\text{adv}}; w), y),\end{aligned} \tag{11}$$

where $x_{\text{adv}} = \arg\min_{\|x'-x\|\leq\delta_{\max}} \ell(f(x'; w), y)$. Here the function $f(x; w)$ is just the classifier branch of discriminator $D(x)$, recalling that we are dealing with conditional GAN. As we can see, throughout the fine-tuning stage, we force the discriminator to focus on the classification task rather than the discrimination task. The experiments will show that the fine-tuning step improves the accuracy by a large margin.

4. Experimental Results

We experiment on both CIFAR10 and a subset of ImageNet data. Specifically, we extract classes y_i such that $y_i \in$ `np.arange(151, 294, 1)` from the original ImageNet data: recall in total there are 1000 classes in ImageNet data and we sampled $294 - 151 = 143$ classes from them. We choose these datasets because 1) the current state-of-the-art GAN, SN-GAN [27], also worked on these datasets, and 2) the current state-of-the-art adversarial training method [23] cannot scale to ImageNet-1k data. In order to have a fair comparison, we copy all the network architectures of generators and discriminators from SN-GAN. Other important factors, such as learning rate, optimization algorithms, and number of discriminator updates in each cycle are also kept the same. The only modification is that we discarded the feature projection layer and applied the auxiliary classifier (see Fig. 4).

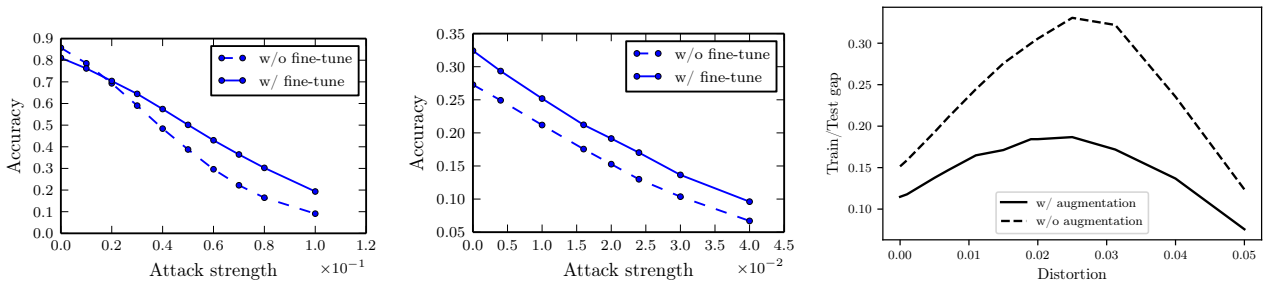

Figure 6: *Left two panels*: The effect of fine-tuning on prediction accuracy (*left:* CIFAR10, *middle:* ImageNet-64px). *Right panel*: Comparing the accuracy gap between adversarial training model and GAN data augmentation model.

4.1. Quality of Discriminator

We show that `Rob-GAN` leads to more robust discriminator than state-of-the-art adversarial trained models.

Effect of fine-tuning. We first compare `Rob-GAN` with/without fine-tuning to verify our claim in Sec. 3.3 that fine-tuning improves classification accuracy. To this end, we compare two sets of models: in the first set, we directly extract the auxiliary classifiers from discriminators to classify images; in the second set, we apply fine-tuning strategy to the pretrained model as Fig. 5 illustrated. The results are in Fig. 6 (left), which suggests that fine-tuning is useful.

Accuracy gap comparison: with or without data augmentation. We check whether adversarial training with fake data augmentation (7) really shrinks the generalization gap. To this end, we draw the same figure as Fig. 2, except that now the classification model is the discriminator of Rob-GAN with fine tuning. We compare the accuracy gap in Fig. 6 (*Right*). Clearly the model trained with the adversarial *real+fake augmentation* strategy works extremely well: it improves the testing accuracy under PGD-attack and so the generalization gap between training/testing set does not increase that much.

Dataset	Defense	$\delta_{\max}$ of ℓ_∞ attacks			
		0	0.02	0.04	0.08
CIFAR10	Adv. training	**81.45%**	69.15%	53.74%	23.58%
	`Rob-GAN` (w/ FT)	81.1%	**70.41%**	**57.43%**	**30.25%**
		0	0.01	0.02	0.03
ImageNet† (64px)	Adv. Training	20.05%	18.3%	12.52%	8.32%
	`Rob-GAN` (w/ FT)	**32.4%**	**25.2%**	**19.1%**	**13.7%**

†Denotes the 143-class subset of ImageNet.

Table 1: Accuracy of our model under ℓ_∞ PGD-attack. "FT" means fine-tuning.

Robustness of discriminator: comparing robustness with/ without data augmentation. In this experiment, we compare the robustness of discriminators trained by Rob-GAN with the state-of-the-art adversarial training algorithm by [23]. As shown in a recent comparison [4], adversarial training [23] achieve state-of-the-art performance in terms of robustness under adversarial attacks. Since adversarial training is equivalent to Rob-GAN without the GAN component, for fair comparison we keep all the other components (network structures) the same.

To test the robustness of different models, we choose the widely used ℓ_∞ PGD attack [23], but other gradient based attacks are expected to yield the same results. We set the ℓ_∞ perturbation to $\delta_{\max} \in$ `np.arange(0, 0.1, 0.01)` as defined in (2). Another minor detail is that we scale the images to $[-1, 1]$ rather than usual $[0, 1]$. This is because generators always have a $\tanh()$ output layer, so we need to do some adaptations accordingly. We present the results in Tab. 1, which clearly shows that our method (`Rob-GAN` w/ FT) performs better than state-of-the-art defense algorithm.

4.2. Quality of Generator

Next we show that by introducing adversarial attack in GAN training, `Rob-GAN` improves the convergence of the generator.

Effect of split classification loss. Here we show the effect of split classification loss described in (10). Recall that if we apply the loss in (9) then the resulting model is AC-GAN. It is known that AC-GAN can easily lose modes in practice, i.e. the generator simply ignores the noise input z and produces fixed images according to the label y. This defect is observed in many previous works [17, 24, 19]. In this ablation experiment, we compare the generated images trained by two loss functions in Fig. 7. Clearly the proposed new loss outperforms the AC-GAN loss.

Quality of generator and convergence speed. Finally, we evaluate the quality of generators trained on two datasets: ImageNet subset - 64px and ImageNet subset - 128px. We compare with the generator obtained by SN-GAN, which has been recognized as a state-of-the-art conditional-GAN model for learning hundreds of classes. Note that SN-GAN can also learn the conditional distribution of the entire ImageNet data (1000 classes), unfortunately, we are not able to match this experiment due to time and hardware limit. To show the performance with/without adversarial training and

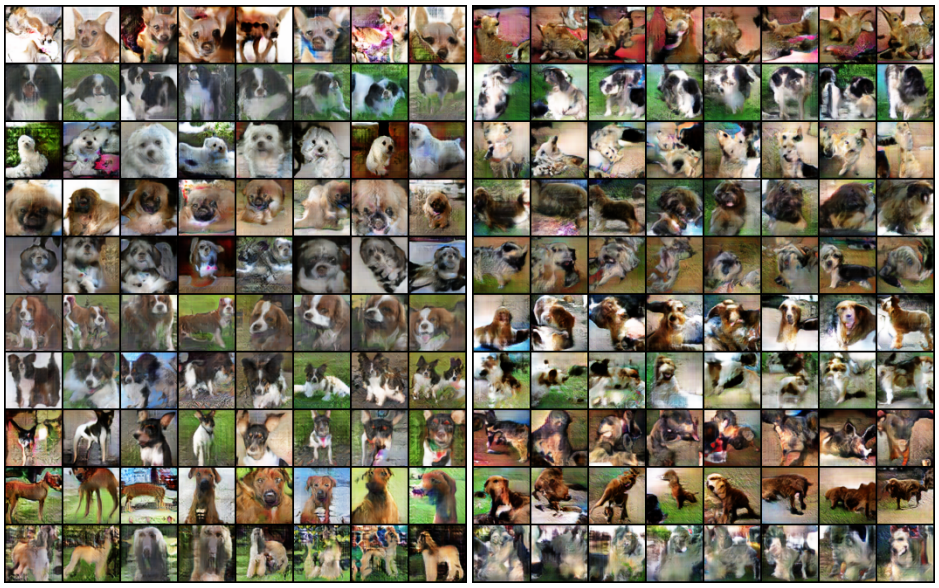

Figure 7: Comparing the generated images trained by our modified loss(*left*) with the original AC-GAN loss(*right*). For fair comparison, both networks are trained by inserting adversarial attacker (Sec. 3.2). We can see images from AC-GAN loss are more distorted and harder to distinguish.

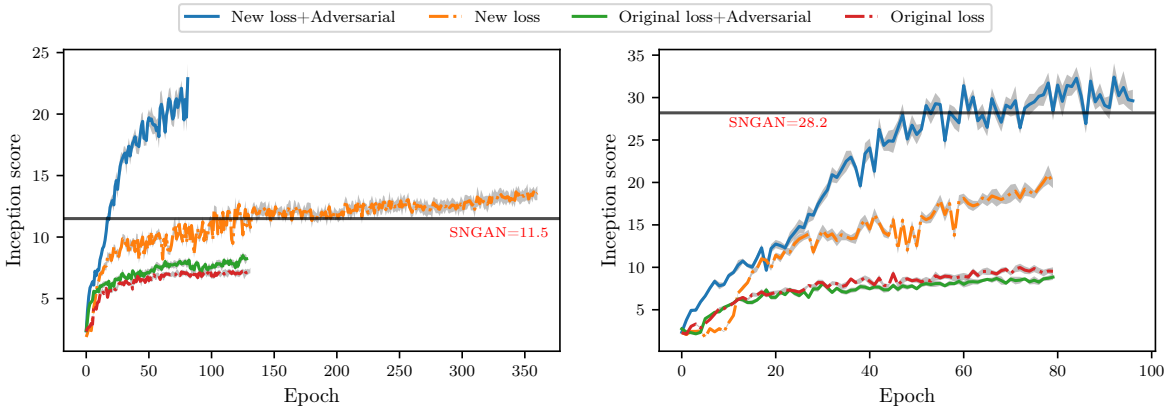

Figure 8: Results on subset of ImageNet, left: 64px, right: 128px. Here we tried four combinations in total: with or without adversarial training, new loss or original loss. We have three findings: 1) Compared with SN-GAN, our model (new loss + adversarial) learns a high quality generator efficiently: in both datasets, our model surpasses SN-GAN in just 25 epochs (64px) or 50 epochs (128px). 2) When comparing the new loss with the original loss, we see the new loss performs better. 3) Using the new loss, the adversarial training algorithm has a great acceleration effect.

with/without new loss, we report the performance of all the four combinations in Figure 8. Note that "original loss" is equivalent to AC-GAN. Based on Figure 8 we can make the following three observations. First, adversarial training can improve the convergence speed of GAN training and make it converge to a better solution. Second, the new loss leads to better solutions on both datasets. Finally, the proposed Rob-GAN outperforms SN-GAN (in terms of inception score) on these two datasets.

5. Conclusions

We show the generator can improve adversarial training, and the adversarial attacker can improve GAN training. Based on these two insights, we proposed to combine generator, discriminator and adversarial attacker in the same system and conduct end-to-end training. The proposed system simultaneously leads to a better generator and a more robust discriminator compared with state-of-the-art models.

Acknowledgments

We acknowledge the support by NSF IIS1719097, Intel, Google Cloud and AITRICS.

References

[1] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017. 2

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 2, 4

[3] A. Athalye and N. Carlini. On the robustness of the cvpr 2018 white-box adversarial example defenses. *arXiv preprint arXiv:1804.03286*, 2018. 3

[4] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018. 2, 3, 7

[5] D. Berthelot, T. Schumm, and L. Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017. 2

[6] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018. 3

[7] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE, 2017. 2

[8] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pages 854–863, 2017. 3

[9] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015. 2

[10] G. S. Dhillon, K. Azizzadenesheli, J. D. Bernstein, J. Kossaifi, A. Khanna, Z. C. Lipton, and A. Anandkumar. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*, 2018. 3

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 4, 6

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2

[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017. 2

[14] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017. 4

[15] C. Guo, M. Rana, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018. 3

[16] A. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009. 6

[17] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie. Stacked generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 4, 2017. 7

[18] H. Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018. 6

[19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017. 7

[20] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 6

[22] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, M. E. Houle, D. Song, and J. Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations*, 2018. 3

[23] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1, 2, 3, 6, 7

[24] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 7

[25] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 5

[26] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. 2, 4, 5

[27] T. Miyato and M. Koyama. cGANs with projection discriminator. In *International Conference on Learning Representations*, 2018. 5, 6

[28] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning*, pages 2642–2651, 2017. 1, 4, 5, 6

[29] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 582–597. IEEE, 2016. 3

[30] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[31] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018. 3

[32] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018. 3

[33] A. Sinha, H. Namkoong, and J. Duchi. Certifiable distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017. 4

[34] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018. 3

[35] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 2

[36] Y. Tokozume, Y. Ushiku, and T. Harada. Learning from between-class examples for deep sound recognition. *arXiv preprint arXiv:1711.10282*, 2017. 6

[37] T. Unterthiner, B. Nessler, G. Klambauer, M. Heusel, H. Ramsauer, and S. Hochreiter. Coulomb gans: Provably optimal nash equilibria via potential fields. *arXiv preprint arXiv:1708.08819*, 2017. 2

[38] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018. 3

[39] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018. 3

[40] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 6